Bonding - Link Aggregation

Linux bonding provides a method for aggregating multiple network interfaces (slaves) into a single logical bonded interface (bond). Cumulus Linux supports two bonding modes:

- IEEE 802.3ad link aggregation mode, which allows one or more links to be aggregated together to form a link aggregation group (LAG), so that a media access control (MAC) client can treat the link aggregation group as if it were a single link. IEEE 802.3ad link aggregation is the default mode.
- Balance-xor mode, where the bonding of slave interfaces are static and all slave interfaces are active for load balancing and fault tolerance purposes. This is useful for MLAG deployments.

The benefits of link aggregation include:

- Linear scaling of bandwidth as links are added to LAG
- Load balancing
- Failover protection

Cumulus Linux uses version 1 of the LAG control protocol (LACP).

To temporarily bring up a bond even when there is no LACP partner, use LACP Bypass.

Contents

- Hash Distribution
- Create a Bond
  - Configuration Options
  - Enable balance-xor Mode
- Example Configuration: Bonding 4 Slaves
- Caveats and Errata
- Related Information

Hash Distribution

Egress traffic through a bond is distributed to a slave based on a packet hash calculation, providing load balancing over the slaves; many conversation flows are distributed over all available slaves to load balance the total traffic. Traffic for a single conversation flow always hashes to the same slave.

The hash calculation uses packet header data to choose to which slave to transmit the packet:

- For IP traffic, IP header source and destination fields are used in the calculation.
- For IP + TCP/UDP traffic, source and destination ports are included in the hash calculation.

In a failover event, the hash calculation is adjusted to steer traffic over available slaves.

Create a Bond

You can create and configure a bond with the Network Command Line Utility (NCLU). Follow the steps below to create a new bond:

1. SSH into the switch.
2. Add a bond using the net add bond command, replacing [bond-name] with the name of the bond, and [slaves] with the list of slaves:

   ```bash
   cumulus@switch:~$ net add bond [bond-name] bond slaves [slaves]
   cumulus@switch:~$ net pending
   cumulus@switch:~$ net commit
   ```

   The bond is configured by default in IEEE 802.3ad link aggregation mode. To configure the bond in balance-xor mode, see bond mode below.
The name of the bond must be:

- Compliant with Linux interface naming conventions.
- Unique within the switch.

## Configuration Options

The configuration options and their default values are listed in the table below.

Each bond configuration option, except for `bond slaves`, is set to the recommended value by default in Cumulus Linux. Only configure an option if a different setting is needed. For more information on configuration values, refer to the `Related Information` section below.

<table>
<thead>
<tr>
<th>NCLU Configuration Option</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>bond mode</td>
<td>The bonding mode. Cumulus Linux supports IEEE 802.3ad link aggregation mode and balance-xor mode. IEEE 802.3ad link aggregation is the default mode. You can change the bond mode using NCLU. The following example changes bond1 to balance-xor mode. <strong>Note</strong>: Use balance-xor mode only if you cannot use LACP. See below for more information.</td>
<td>802.3ad</td>
</tr>
<tr>
<td></td>
<td>cumulus@switch:~$ net add bond bond1 bond mode balance-xor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cumulus@switch:~$ net pending</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cumulus@switch:~$ net commit</td>
<td></td>
</tr>
</tbody>
</table>

The following example changes bond1 to IEEE 802.3ad link aggregation mode:

```bash
cumulus@switch:~$ net add bond bond1 bond mode 802.3ad
cumulus@switch:~$ net pending
cumulus@switch:~$ net commit
```

<table>
<thead>
<tr>
<th>bond slaves</th>
<th>The list of slaves in the bond.</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>bond miimon</td>
<td>Defines how often the link state of each slave is inspected for failures.</td>
<td>100</td>
</tr>
<tr>
<td>bond downdelay</td>
<td>Specifies the time, in milliseconds, to wait before disabling a slave after a link failure has been detected. This option is only valid for the miimon link monitor. The downdelay value must be a multiple of the miimon value; if not, it is rounded down to the nearest multiple.</td>
<td>0</td>
</tr>
<tr>
<td>bond updelay</td>
<td>Specifies the time, in milliseconds, to wait before enabling a slave after a link recovery has been detected. This option is only valid for the miimon link monitor. The updelay value must be a multiple of the miimon value; if not, it is rounded down to the nearest multiple.</td>
<td>0</td>
</tr>
<tr>
<td>bond use-carrier</td>
<td>Determines the link state.</td>
<td>1</td>
</tr>
<tr>
<td>bond xmit-hash-policy</td>
<td>The hash method used to select the slave for a given packet.</td>
<td>layer3+4</td>
</tr>
<tr>
<td></td>
<td>Do not change this setting.</td>
<td></td>
</tr>
<tr>
<td>bond lacp-bypass-allow</td>
<td>Enables LACP bypass.</td>
<td>N/A</td>
</tr>
<tr>
<td>bond lACP-rate</td>
<td>Sets the rate to ask the link partner to transmit LACP control packets.</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You can set the LACP rate to slow using NCLU:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>```</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cumulus@switch:~$ net add bond bond01 bond</td>
<td></td>
</tr>
<tr>
<td></td>
<td>lACP-rate slow</td>
<td></td>
</tr>
<tr>
<td>bond min-links</td>
<td>Defines the minimum number of links that must be active before the bond is put into service.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A value greater than 1 is useful if higher level services need to ensure a minimum aggregate bandwidth level before activating a bond. Keeping bond-min-links set to 1 indicates the bond must have at least one active member. If the number of active members drops below the bond-min-links setting, the bond will appear to upper-level protocols as link-down. When the number of active links returns to greater than or equal to bond-min-links, the bond becomes link-up.</td>
<td></td>
</tr>
</tbody>
</table>

**Enable balance-xor Mode**

When you enable *balance-xor mode*, the bonding of slave interfaces are static and all slave interfaces are active for load balancing and fault tolerance purposes. Packet transmission on the bond is based on the hash policy specified by `xmit-hash-policy`.

When using balance-xor mode to dual-connect host-facing bonds in an MLAG environment, you **must** configure the `clag_id` parameter on the MLAG bonds and it must be the same on both MLAG switches. Otherwise, the bonds are treated by the MLAG switch pair as single-connected.

Use *balance-xor mode* only if you cannot use LACP; LACP can detect mismatched link attributes between bond members and can even detect misconnections.

To change the mode of an existing bond to balance-xor, run the `net add bond <bond-name> bond mode balance-xor` command. The following example commands change bond1 to balance-xor mode:

```bash
cumulus@switch:~$ net add bond bond1 bond mode balance-xor
cumulus@switch:~$ net pending
cumulus@switch:~$ net commit
```

To create a new bond and configure the bond to use balance-xor mode, create the bond, then configure the bond mode. The following example commands create a bond called bond1 and configure bond mode to be balance-xor:

```bash
cumulus@switch:~$ net add bond bond1 bond slaves swp3,4
cumulus@switch:~$ net add bond bond1 bond mode balance-xor
cumulus@switch:~$ net pending
cumulus@switch:~$ net commit
```

These commands create the following configuration in the `/etc/network/interfaces` file:

```bash
auto bond1
iface bond1
    bond-mode balance-xor
    bond-slaves swp3 swp4
```

To view the bond, use NCLU:
cumulus@switch:~$ net show interface bond1

<table>
<thead>
<tr>
<th>Name</th>
<th>MAC</th>
<th>Speed</th>
<th>MTU</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>UP</td>
<td>bond1</td>
<td>00:02:00:00:00:12</td>
<td>20G</td>
<td>1500</td>
</tr>
</tbody>
</table>

Bond Details

---

Bond Mode: Balance-XOR
Load Balancing: Layer3+4
Minimum Links: 1
In CLAG: CLAG Inactive

<table>
<thead>
<tr>
<th>Port</th>
<th>Speed</th>
<th>TX</th>
<th>RX</th>
<th>Err</th>
<th>Link Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>UP</td>
<td>10G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>UP</td>
<td>10G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

LLDP

---

swp3(P) ==== swp1(plch1)
swp4(P) ==== swp2(plch1)Routing

---

Interface bond1 is up, line protocol is up
Link ups: 3 last: 2017/04/26 21:00:38.26
Link downs: 2 last: 2017/04/26 20:59:56.78
PTM status: disabled
vrf: Default-IP-Routing-Table
index 31 metric 0 mtu 1500
flags: <UP,BROADCAST,RUNNING,MULTICAST>
Type: Ethernet
HWaddr: 00:02:00:00:00:12
inet6 fe80::202:ff:fe00:12/64
Interface Type Other

Example Configuration: Bonding 4 Slaves

In the following example, the front panel port interfaces swp1 thru swp4 are slaves in bond0, while swp5 and swp6 are not part of bond0.
When networking is started on the switch, bond0 is created as MASTER and interfaces swp1 thru swp4 come up in SLAVE mode, as seen in the `ip link show` command:

**Example Bond Configuration**

The following commands create a bond with four slaves:

```bash
cumulus@switch:~$ net add bond bond0 address 10.0.0.1/30
cumulus@switch:~$ net add bond bond0 bond slaves swp1-4
cumulus@switch:~$ net pending
cumulus@switch:~$ net commit
```

These commands create this code snippet in the `/etc/network/interfaces` file:

```bash
auto bond0
iface bond0
    address 10.0.0.1/30
    bond-slaves swp1 swp2 swp3 swp4
```

If the bond is going to become part of a bridge, you do not need to specify an IP address.

When networking is started on the switch, bond0 is created as MASTER and interfaces swp1 thru swp4 come up in SLAVE mode, as seen in the `ip link show` command:
cumulus@switch:~$ ip link show
...
3: swp1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
   master bond0 state UP mode DEFAULT qlen 500
   link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
4: swp2: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
   master bond0 state UP mode DEFAULT qlen 500
   link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
5: swp3: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
   master bond0 state UP mode DEFAULT qlen 500
   link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
6: swp4: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
   master bond0 state UP mode DEFAULT qlen 500
   link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
...
55: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue
   state UP mode DEFAULT
   link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff

All slave interfaces within a bond have the same MAC address as the bond. Typically, the first slave added to the bond donates its MAC address as the bond MAC address, whereas the MAC addresses of the other slaves are set to the bond MAC address.

The bond MAC address is used as the source MAC address for all traffic leaving the bond and provides a single destination MAC address to address traffic to the bond.

Caveats and Errata

- An interface cannot belong to multiple bonds.
- A bond can have subinterfaces, but subinterfaces cannot have a bond.
- A bond cannot enslave VLAN subinterfaces.
- Set all slave ports within a bond to the same speed/duplex and make sure they match the link partner’s slave ports.
- On a Cumulus RMP switch, if you create a bond with multiple 10G member ports, traffic gets dropped when the bond uses members of the same unit listed in the /var/lib/cumulus/porttab file. For example, traffic gets dropped if both swp49 and swp52 are in the bond because they both are in the xe0 unit (or if both swp50 and swp51 are in the same bond because they are both in xe1):
  swp49 xe0 0 0 -1 0
  swp50 xe1 0 0 -1 0
  swp51 xe1 1 0 -1 0
  swp52 xe0 1 0 -1 0
Single port member bonds, bonds with different units (xe0 or xe1, as above), or layer 3 bonds do not have this issue.

On Cumulus RMP switches, which are built with two Hurricane2 ASICs, you cannot form an LACP bond on links that terminate on different Hurricane2 ASICs.

Related Information

- Linux Foundation - Bonding
- 802.3ad (Accessible writeup)
- Wikipedia - Link aggregation